WHAT IS CLAIMED IS:

1. A method for operating a fault-tolerant server group in client-server distributed dynamic network systems, comprising:

receiving, by a master server in a fault-tolerant server group, a request sent by a client, said fault-tolerant server group comprising said master server and at least one back-up server, said master server communicating with both said client and said at least one back-up server, each server in said server group, including said master server and said at least one back-up server, having an election mechanism enabling said fault-tolerant server group to elect a new master server , when said master server fails, in a process in which at least some of the election mechanisms are triggered at different times;

processing, by said fault-tolerant server group, said request to produce a result, said request being processed concurrently by said master server and said at least one back-up server; and

sending, by said master server, said result to said client.

2. The method according to claim 1, further comprising

determining, by said election mechanism running on said at least one back-up server, whether said master server is functional; and

electing a new master server for said fault-tolerant server group when said master server fails, said electing being performed based on at least one election periodic timer associated with said at least one back-up server, said new master server assuming said mastership and communicating with both said client and other servers in said fault-tolerant server group.

3. A method for operating an election mechanism, comprising:

detecting, by at least one back-up server, when a master server is not functional, said master server possessing the mastership of a server group, said server group comprising said master server and said at least one back-up server; and

electing, when said master server is detected not functional, a new master server based on at least one election periodic timer, each of said at least one election periodic timer being associated with a different server in said server group and causing said electing to occur at different times for at least some of the servers, said new master server taking over said mastership.

4. The method according to claim 3, wherein said detecting comprises:

sending, by each of said at least one back-up server, at least one inquiry message to said master server;

examining, by said at least one back-up server, whether a reply, from said master server as a response to said at least one inquiry message, is received within a certain amount of time;

determining that said master server is not functional if said reply is not received within said certain amount of time; and

determining that said master server is functional if said reply is received from said master server within said certain amount of time.

5. The method according to claim 4, further comprising:

initializing a time-out mechanism after said sending, said time-out mechanism including a time-out condition that defines said certain amoun of time, said time-out mechanism starting to count time towards said time-out condition after said initializing.

6. The method according to claim 3, wherein said electing comprises:

selecting one of said at least one back-up server as the new master server, said selecting being carried out based on said at least one election periodic timer; and

updating said mastership based on said new master server.

7. The method according to claim 6, wherein said selecting comprises:

setting the state of a back-up server, whose master server is detected not functional, to be a waiting state;

initializing an election periodic timer associated with said back-up server, said election periodic timer specifying a particular length of time defined for said back-up server; and

declaring, by said back-up server, said back-up server as the new master server, if no message from a different server is received during said particular length of time.

8. The method according to claim 7, wherein said declaring comprises:

setting the state of said back-up server to be master; and

sending a first message to said at least one back-up server and said master server, said first message declaring said back-up server as the new master server.

9. The method according to claim 7, wherein said particular length of time specified by said election periodic timer is an election delay time associated with said back-up server.

10. The method according to claim 9, wherein said election delay time associated with said back-up server is determined by a summation of a base election delay time and an accumulative maximum communication delays between each of the back-up servers in a first set of back-up servers to a second set of back-up servers.

11. The method according to claim 10, wherein said first set of back-up servers corresponds to the back-up servers in said server group that have smaller index than the index of said back-up server.

12. The method according to claim 10, wherein said second set of back-up servers includes the back-up servers in said server group that do not have smaller index than the index of said back-up server.

13. The method according to claim 10, wherein each of said maximum communication delays is computed as the maximum value in communication delay between a back-up server from said first set of back-up servers and all the back-up servers from said second set of back-up servers.

14. The method according to claim 8, wherein

said declaring further comprising

receiving a message from a different server after said sending; and

said detecting comprises:

receiving a message from a different server after said master server is detected functional.

15. The method according to claim 14, wherein said updating comprises:

analyzing, by a receiving back-up server that receives said message from a different server, said message to determine whether said message is a first message or a second message, said first message declares said mastership by said new master server, said second messsage requesting said receiving back-up server to replace said mastership with the mastership declared by the new master server that sends said second message;

establishing, if said message is a first message, said mastership based on a new master server declared in said first message; and

determining, if said message is a second message, whether to accept said mastership declared by said new master server in said second message.

16. The method according to claim 15, wherein said establishing comprises:

acknowledging, when the state of said receiving back-up server is either a waiting state or a back-up state, said mastership declared by said new master server in said first message; and

determining, when the state of said receiving back-up server is neither a waiting state nor a back-up state, whether to accept said mastership declared by said new master server in said first message.

17. The method according to claim 16, wherein said acknowledging comprises:

setting the master state of said receiving back-up server to be said new master server; and

terminating said election periodic timer associated with said receiving back-up server.

18. The method according to claim 16, wherein said determining comprises:

comparing said receiving back-up server with said new master server according to at least one criterion to yield a result of either positive, indicating that said receiving back-up server wins, or negative, indicating that said receiving back-up server loses;

accepting said mastership, declared by said new master server, if said result is negative; and

revoking said mastership, declared by said new master server, if said result is positive.

19.   The method according to claim 18, wherein said at least one criterion includes that the index of said new master server is greater than the index of said receiving back-up server.

20.   The method according to claim 19, wherein said index of a server is determined according to the computational power of said server.

21.   The method according to claim 19, wherein said index of a server is determined according to the bandwidth capacity of said server.

22.   The method according to claim 18, wherein said accepting comprises:

setting the state of said receiving back-up server to be back-up; and

setting the master state of said receiving back-up server to be said new master server.

23.   The method according to claim 18, wherein said revoking comprises:

setting the state of said receiving back-up server to be master; and

sending, by said receiving back-up server, a second message to all other servers in said server group, said second message carrying both the index of said new master server and the index of said receiving back-up server and requesting said all other servers to revoke the mastership previously declared by said new master server, represented by the index of said new master server, and to accept the mastership currently declared by said receiving back-up server, represented by said index of said receiving back-up server.

24.   The method according to claim 3, further comprising triggering said detecting based on a detection periodic timer.

25.   The method according to claim 24 wherein said detection periodic timer includes an equally periodic timer.

26.   A fault-tolerant server group in distributed dynamic network systems, comprising:

a client; and

a fault-tolerant server group for providing a service to said client, said fault-tolerant server group comprising a master server and at least one back-up server, said master server communicating with said client, wherein each server in said fault-tolerant server group, including said master server and said back-up server, is associated with an election mechanism enabling said fault-tolerant server group to elect, when said master server fails, a new master server, in a process in which at least some of the election mechanisms are triggered at different times.

27.   The server group according to claim 26, further comprising a name server for registering the mastership of said master server.

28.   A system for an election mechanism, comprising:

a detection mchanism for detecting, by at least one back-up server, when a master server is not functional, said master server possessing the mastership of a fault-tolerant server group, said server group comprising said master server and said at least one back-up server; and

an election mechanism for electing, when said master server is detected not functional, a new master server based on at least one election periodic timer, each of said at least one election periodic timer being associated with a different server in said server group and causing at least some of the servers to elect at different times, said new master server taking over said mastership.

29.   The system according to claim 28, wherein said election mechanism comprises:

an election periodic timer associated with a back-up server for controlling the election delay time of said back-up server;

a master selection mechanismfor selecting said new master server based on said election periodic timer; and

a mastership updating mechanism for updating said mastership based on said new master server.

30.   A computer readable medium having program code stored thereon, such that when the code is read and executed by a computer, the computer is caused to:

receive, by a master server in a fault-tolerant server group, a request sent by a client, said fault-tolerant server group comprising said master server and at least one back-up server, said master server communicating with both said client and said at least one back-up server, each server in said server group, including said master server and said at least one back-up server, having an election mechanism, enabling said fault-tolerant server group to elect, when said master server fails, a new master server, in a process in which at least some of the election mechanisms are triggered at different times;

process, by said fault-tolerant server group, said request to produce a result, said request being processed concurrently by said master server and said at least one back-up server; and

send, by said master server, said result to said client.

31.   The medium according to claim 30, wherein the code recorded on the medium further causes said computer to:

determine, by said election mechanism runnin on said at least one back-up server, whether said master is functional; and

elect a new master server for said fault-tolerant server group, when said master server fails, said electing being performed based on at least one election periodic timer associated with said at least one back-up server, said new master server assuming said mastership and communicating with both said client and other servers in said fault-tolerant server group.

32. A computer readable medium having program code stored thereon, such that when the code is read and executed by a computer, the computer is caused to:

detect, by at least one back-up server, when a master server is not functional, said master server possessing the mastership of a server group, said server group comprising said master server and said at least one back-up server; and

elect, when said master server is detected not functional, a new master server based on at least one election periodic timer, each of said at least one election periodic timer being associated with a different server in said server group and causing at least some of the servers to elect at different times, said new master server taking over said mastership.

33. The medium according to claim 32, wherein the code recorded on the medium further causes said computer to:

send, by each of said at least one back-up server, at least one inquiry message to said master server;

examine, by said at least one back-up server, whether a reply, from said master server as a response to said at least one inquiry message, is received within a certain amount of time;

determine that said master server is not functional if said reply is not received within said certain amount of time; and

determine that said master server is functional if said reply is received from said master server within said certain amount of time.

34.   The medium according to claim 33, wherein the code recorded on the medium further causes said computer to:

initialize a time-out mechanism after said send that includes a time-out condition that defines said certain amount of time, said time-out mechanism starting to count time towards said time-out condition after said initialize.

35.   The medium according to claim 32, wherein the code recorded on the medium further causes said computer to:

select one of said at least one back-up server as the new master server, said selecting being carried out based on said at least one election periodic timer; and

update said mastership based on said new master server, selected by said selecting.

36. The medium according to claim 35, wherein the code recorded on the medium further causes said computer to:

set the state of a back-up server, whose master server is detected not functional, to be a waiting state;

initialize an election periodic timer associated with said back-up server, said election periodic timer specifying a particular length of time defined for said back-up server; and

declare, by said back-up server, said back-up server as the new master server, if no message from a different server is received during said particular length of time.

37. The medium according to claim 36, wherein the code recorded on the medium further causes said computer to:

set the state of said back-up server to be master; and

send a first message to said at least one back-up server and said master server, said first message declaring said back-up server as the new master server.

38. The medium according to claim 36, wherein said declare further comprising:

receiving a message from a different server after said send; and

said detect comprises:

receiving a message from a different server after said master server is detected functional.

39. The medium according to claim 38, wherein the code recorded on the medium further causes said computer to:

analyze, by a receiving back-up server that receives said message from a different server, said message to determin whether said message is a first message or a second message, said first message declares said mastership by said new master server, said second messsage requesting said receiving back-up server to replace said mastership with the mastership declared by the new master server that sends said second message;

establish, if said message is a first message, said mastership based on a new master server declared in said first message, said new master server being represented in said first message by an index designated to said new master server; and

determine, if said message is a second message, whether to accept said mastership declared by said new master server in said second message.

40. The medium according to claim 39, wherein the code recorded on the medium further causes said computer to:

acknowledge, when the state of said receiving back-up server is either a waiting state or a back-up state, said mastership declared by said new master server in said first message; and

determine, when the state of said receiving back-up server is neither the waiting state nor the back-up state, whether to accept said mastership declared by said new master server in said first message.

41. The medium according to claim 40, wherein the code recorded on the medium further causes said computer to:

set the master state of said receiving back-up server to be said new master server; and

terminate said election periodic timer associated with said receiving back-up server.

42. The medium according to claim 40, wherein the code recorded on the medium further causes said computer to:

compare said receiving back-up server with said new master server according to at least one criterion to yield a result of either positive, indicating that said receiving back-up server wins, or negative, indicating that said receiving back-up server loses;

accept said mastership, declared by said new master server, if said result is negative; and

revoke said mastership, declared by said new master server, if said result is positive.

43. The medium according to claim 42, wherein said at least one criterion includes that the index designated to said new master server is greater than the index designated to said receiving back-up server.

44. The medium according to claim 42, wherein said accept comprises:

setting the state of said receiving back-up server to be back-up; and

setting the master state of said receiving back-up server to be said new master server.

45. The medium according to claim 42, wherein said revoke comprises:

setting the state of said receiving back-up server to be master; and

sending, by said receiving back-up server, a second message to all other servers in said server group, said second message carrying both the index of said new master server and the index of said receiving back-up server and requesting said all other servers to revoke the mastership previously declared by said new master server, represented by the index of said new master server, and to accept the mastership currently declared by said receiving back-up server, represented by said index of said receiving back-up server.

46. The medium according to claim 32, wherein the code recorded on the medium further causes said computer to trigger said detecting based on a detection periodic timer.

47. The medium according to claim 46, wherein said detection periodic timer includes an equally periodic timer.